

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

Elektronická mobilní kuchařka  
s přizpůsobením receptů  
Mobile Cookbook with Automatic  
Cooking Recipe Adjustment

2017

Erika Čížová

# Zadání bakalářské práce

Student: **Erika Čížová**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Elektronická mobilní kuchařka s přizpůsobením receptů**  
**Mobile Cookbook with Automatic Cooking Recipe Adjustment**

Jazyk vypracování: čeština

## Zásady pro vypracování:

Cílem práce je vytvořit mobilní aplikaci pro správu kuchařských receptů a jejich automatizované úpravy na platformě Android. Aplikace bude umožňovat sdílení receptů a jejich hodnocení, výběr receptů podle dostupných ingrediencí (včetně skenování čárového kódu), export nákupního seznamu, úpravy receptu (podle počtu porcí resp. generování nových receptů) a časování výrobního postupu.

1. Prostudujte existující elektronické aplikace pro správu kuchařských receptů na mobilních platformách.
2. Nastudujte a popište metody, jakými lze provést modifikace receptů.
3. Analyzujte požadavky, které budou kladeny na aplikaci a navrhnete uživatelské rozhraní.
4. Implementujte aplikaci a případnou serverovou část, určenou pro sdílení receptů.
5. Otestujte funkčnost aplikace a vyhodnoťte dosažené výsledky, posuďte použitelnost vygenerovaných upravených receptů.

## Seznam doporučené odborné literatury:

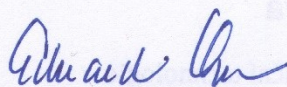
- [1] Steele, J., To, N.: The Android Developer's Cookbook: Building Applications with the Android SDK, Addison-Wesley Professional, 2010, ISBN-13: 978-0321741233.
- [2] Meier, R.: Professional Android 4 Application Development, Wrox, 2012, ISBN-13: 978-1118102275
- [3] Hashimi, S.: Pro Android 2, Apress, 2010, ISBN-13: 978-1430226598.
- [4] Recipe Converter [online] [cit. 2015-09-10] Dostupné z:  
<<http://mykitchencalculator.com/recipeconverter.html>>
- [5] Android Barcode Scanner Library [online] [cit. 2015-09-10] Dostupné z:  
<<https://github.com/dm77/barcodescanner>>

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

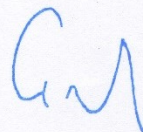
Vedoucí bakalářské práce: **Ing. Pavel Moravec, Ph.D.**

Datum zadání: 01.09.2015

Datum odevzdání: 28.04.2017



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty



## Čestné prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě dne 28. 4. 2017



.....

podpis

## **Poděkování**

Na tomto místě bych ráda poděkovala panu Ing. Pavlu Moravcovi, Ph.D. za odborné vedení práce, cenné rady a vstřícnost při konzultacích a vypracovávání této bakalářské práce.

## **Abstrakt**

Cílem této práce je vytvořit mobilní aplikaci pro správu kuchařských receptů a jejich automatizované úpravy. Tato aplikace umožňuje sdílení receptů spolu s jejich hodnocením, výběr receptů podle dostupných ingrediencí a úpravy receptů podle počtu porcí. V aplikaci je také zabudováno skenování čárových kódů potravin, podle kterých následně lze vyhledávat recepty. Dále má uživatel aplikace možnost vytvoření zcela nového receptu a úpravy již existujících receptů. Aplikace také umožňuje export potravin do editovatelného nákupního seznamu a časování výrobního postupu.

Hlavní přínos této práce spočívá ve sloučení mnoha funkcí z již existujících (i placených) aplikací a také přidání nových funkcionalit, jako je například výše zmíněné skenování potravin nebo možnost spuštění několika na sobě nezávislých časovačů.

Aplikace je implementována v programovacím jazyce Java. První část této práce je věnována popisu existujících aplikací. V dalších kapitolách je popsána analýza požadavků, a to s využitím jazyka UML, spolu s popisem uživatelského rozhraní aplikace. Dále je také popsán postup implementace některých částí aplikace spolu s popisem využití SQLite databáze.

## **Klíčová slova**

Android, Java, Receptář, čárové kódy potravin, skenování, časování výrobního procesu, databáze, sdílení obsahu, hodnocení.

## **Abstract**

The aim of this work is to create mobile application for managing cooking recipes and their automated adjustments. The application allows to share recipes and their rating with others, then to choose recipe according to available ingredients and to adjust recipes according to number of serving. In the application there is also included scanning of barcodes, according to which the user can search recipes. The application offers a way to create a brand new recipe and to adjust those that already have been created. Further more the application enables to export the ingredients to the shopping list and also to add multiple timers for easier timing of manufacturing process.

Main benefit of this application can be founded in combination of many functions from already existing applications (even the paid ones) and also adding new functionalities, for example like was said before the opportunity to scan barcode of ingredients or adding multiple independent timers.

The application is implemented in Java programming language. The first part of this work is dedicated to description of existing applications. In next parts there is analysis of requirements with use of UML diagrams along with description of user interface. Further more there is described implementation of some parts of application along with the usage of SQLite database.

## **Key Words**

Android, Java, CookBook, barcodes of foodstuff, scanning, timing of manufacturing process, database, sharing of content, rating.

## Obsah

1	Úvod.....	12
2	Průzkum existujících aplikací .....	13
2.1	sRecepty .....	13
2.2	Česká kuchařka .....	14
2.3	Miluji Vaření – recepty .....	14
2.4	Vaření.cz - Nejlepší recepty .....	14
3	Metody modifikace receptů.....	17
4	Analýza požadavků kladených na aplikaci .....	18
4.1	Use Case diagram.....	18
4.2	Třídní diagram.....	19
4.3	Sekvenční diagram .....	20
4.4	Návrh databáze – relační datový model .....	22
4.4.1	Kardinalita vztahů entit .....	22
5	Uživatelské rozhraní.....	23
5.1	Hlavní obrazovka .....	23
5.2	Zobrazení detailu receptu .....	24
5.2.1	Karta Přehled.....	24
5.2.2	Karta Suroviny .....	24
5.2.3	Karta Postup .....	25
5.3	Úprava receptu .....	25
5.4	Hodnocení receptu.....	26
5.5	Vytvoření nového receptu a sdílení receptů .....	26
5.6	Vyhledávání receptu podle názvu i podle surovin .....	27
5.7	Spravování databáze EANů .....	28
5.8	Časování výrobního procesu – Časovač.....	29
5.9	Nákupní seznam .....	30
5.10	Zobrazení oblíbených receptů .....	31
5.11	Nastavení aplikace.....	31
6	Implementace Receptáře .....	33
6.1	SQLite databáze .....	33
6.1.1	Využití asynchronního přístupu pro práci s databází .....	33
6.2	Použití aplikace Barcode Scanner .....	34
6.3	Přidání obrázku do receptu a jeho následné zobrazení v detailu receptu .....	34
6.3.1	Výběr obrázku z galerie .....	35



6.3.2	Vyfotografování nového obrázku.....	35
6.3.3	Zobrazení obrázku v aplikaci .....	36
6.4	Nákupní seznam .....	37
6.4.1	Shared Preferences vs. Internal Storage vs. External Storage.....	37
6.4.2	Uložení nákupního seznamu .....	38
6.5	Časovač .....	39
6.5.1	Vzhled seznamu časovačů.....	40
6.5.2	Řešení časování výrobního postupu .....	40
6.5.3	Upozornění - AlarmManager .....	40
6.6	Rozšířené vyhledávání receptů.....	42
6.7	Sdílení receptů.....	42
7	Závěr .....	44
	Reference.....	46
	Seznam příloh.....	48

## **Seznam použitých symbolů a zkratek**

GUI	Graphical User Interface
UI	User Interface
EAN	European Article Number (Čárový kód potravin)
SQL	Structured Query Language
JSON	JavaScript Object Notation
XML	eXtensible Markup Language
UML	Unified Modeling Language

## Seznam ilustrací

Obrázek 1: Use Case diagram funkcionalit aplikace.....	19
Obrázek 2: Třídní diagram, rozšířené vyhledávání .....	20
Obrázek 3: Sekvenční diagram, vyhledávání receptu pomocí naskenování EANu suroviny .....	21
Obrázek 4: Relační databázový model - Recepty.db, EANy.db .....	22
Obrázek 5: Vzhled hlavní obrazovky .....	23
Obrázek 6: Zobrazení detailu receptu .....	25
Obrázek 7: Možnosti sdílení receptu.....	27
Obrázek 8: Sdílení pomocí G-mailu.....	27
Obrázek 9: Rozšířené vyhledávání, zadání suroviny .....	28
Obrázek 10: Rozšířené vyhledávání, skenování EANů .....	28
Obrázek 11: Seznam časovačů.....	30
Obrázek 12: Vytvoření nového časovače.....	30
Obrázek 13: Nákupní seznam .....	31
Obrázek 14: Nastavení aplikace .....	32
Obrázek 15: Zobrazení oblíbených receptů.....	32
Obrázek 16: Položka seznamu .....	40

## Seznam tabulek

Tabulka 1: Srovnání existujících aplikací pro správu kuchařských receptů .....	15
Tabulka 2: Kardinalita vztahů entit v databázi.....	22

## Seznam výpisů zdrojového kódu

Zdrojový kód 1: Nabídka možností přidání obrázku k receptu.....	35
Zdrojový kód 2: Výběr obrázku z galerie .....	35
Zdrojový kód 3: Vyfotografování nového obrázku.....	36
Zdrojový kód 4: Zobrazení obrázku.....	36
Zdrojový kód 5: Zobrazení obrázku na celou obrazovku.....	37
Zdrojový kód 6: Načtení a uložení dat – Shared Preferences .....	39
Zdrojový kód 7: Upozornění ve stavovém řádku .....	41
Zdrojový kód 8: Nastavení možností časovače.....	41
Zdrojový kód 9: Výběr aplikace pro sdílení receptu.....	42
Zdrojový kód 10: Filtrace výběru aplikací pro sdílení receptu .....	43

# 1 Úvod

Obsahem Bakalářské práce je popis implementace aplikace, nazvané Receptář, jejíž hlavním aspektem je možnost vyhledávání receptů podle obsahu spížířny. Funkcionalita je řešena následovně: uživatel má možnost vyhledávat recept podle ručně zadaných ingrediencí, anebo může vyfotit čárový kód produktu/potraviny. Aplikace poté nabídne seznam receptů, které obsahují zadané potraviny nebo alespoň část z nich.

Aplikace podporuje také vyhledávání receptů podle samotného názvu receptu nebo podle kategorie, do které recept logicky spadá.

Uživatel může při prohlížení receptů vybírat suroviny, které nemá ve své spížířně. Tyto suroviny může postupně exportovat do snadno editovatelného nákupního seznamu, který je také součástí aplikace.

Recepty je také možno sdílet s ostatními pomocí různých aplikací, jako je například Messenger od Facebooku nebo G-mail. Ve sdílení jsou zahrnuty veškeré informace, které jsou v receptu uvedeny, spolu s hodnocením a komentářem k danému receptu. Pro sdílení je samozřejmě nutné, aby byl uživatel připojen k Internetu, pro fungování ostatních funkcí aplikace však postačí offline režim.

Dále aplikace obsahuje vlastní minutku neboli časovač. U každého receptu je stanovena doba výrobního postupu a pomocí tlačítka lze zapnout časovač se zadanou dobou získanou přímo z receptu. Časovač zůstává funkční i po zavření aplikace. Uživatel má také možnost spustit několik časovačů zároveň.

Samotné recepty je dále možno jakkoliv editovat. Tím je myšleno jak přepočítávání množství surovin podle počtu porcí, které chce uživatel uvařit, tak také úpravy celého receptu, výrobního postupu i samotných surovin. K receptu je také možné připojit komentář a vlastní hodnocení, stejně jako nastavit recept jako oblíbený.

Aplikace využívá databázi receptů a také databázi kódů potravin (EAN), obě jsou však pouze lokální a tudíž není zapotřebí přístupu k internetu. Toto řešení ovšem odepírá možnost nalezení daného receptu v některé z existujících velkých databází receptů – uživatel má k dispozici pouze ty recepty, které jsou v základním setu databáze anebo ty, které sám vytvoří.

## 2 Průzkum existujících aplikací

Existuje mnoho aplikací, které se zabývají správou kuchařských receptů. Některé jsou placené, jiné jsou zdarma. Často však mají podobnou funkcionalitu a liší se výrazně pouze grafickým zpracováním aplikace a samozřejmě obsahem databáze receptů.

Narážíme zde však na jazykový problém. U velké části aplikací pro správu receptů je obsah databáze receptů psán jen v anglickém jazyce anebo mají zvláštní, ba i pochybný překlad do češtiny, způsobený automatizovaným překladačem. Některé aplikace se dokonce „tváří“ jako české, ale týká se to pouze překladu UI, nikoli překladu samotných receptů (tedy databáze receptů je stále v původním, originálním, jazyce). Samozřejmě pokud uživatel hledá aplikace pouze v češtině, má na výběr z poměrně velkého množství aplikací.

Jiné aplikace jsou zdánlivě v naprostém pořádku, avšak vlivem špatné implementace jsou pomalé a jejich načítání přesahuje přiměřenou dobu čekání uživatele na odezvu. Častým komentářem v Obchodu Play [1] bylo také to, že aplikace je po nainstalování nestabilní a někdy ji ani nelze spustit.

Dalším aspektem aplikací pro správu kuchařských receptů je také možnost práce v tzv. offline režimu. Mezi nalezenými aplikacemi se vyskytovaly i takové, které sice poskytovaly uživateli GUI, ale bez připojení k Internetu se bohužel na žádný recept nepodívali. V jiných případech docházelo při prvním spuštění aplikace ke stažení velkého souboru původně online databáze do paměti telefonu, což mohlo u velké části uživatelů způsobit problémy.

Dále pak máme aplikace, které mají velmi neintuitivní uživatelské rozhraní, a uživatel z nich dostává pocit neorganizovanosti a je využito zbytečně složité navigace mezi funkcionalitami aplikace. Takovéto aplikace však mají velmi slabé hodnocení, takže se k nim běžný uživatel, který hledá aplikaci pro správu kuchařských receptů, ani nedostane.

Níže jsou pečlivě vybrány a popsány aplikace, které splňují velkou část požadavků běžného uživatele. Následně je připojena tabulka pro celkové srovnání těchto aplikací s implementovanou aplikací Receptář.

### 2.1 sRecepty



Aplikace sRecepty [2] je podle popisu na Obchodu Play zcela online aplikace, která obsahuje obrovskou databázi receptů. Podle uživatelů však aplikace neobsahuje možnost přidání vlastního receptu a recepty obsažené v databázi často neodpovídají požadavkům běžného uživatele. Co je také ne úplně správně vyřešeno, je možnost sdílení receptu s přáteli. Podle komentářů uživatelů je při sdílení zaslán pouze odkaz na webovou stránku, kde se recept nachází, přičemž uživatelé by určitě uvítali spíš možnost zaslat rovnou celý obsah receptu.

## 2.2 Česká kuchařka



Placená aplikace, která ačkoliv má vysoké hodnocení, tak podle komentářů od uživatelů má daleko k dokonalosti. Ač GUI aplikace je velmi povedené, bohužel databáze receptů je velmi neobsáhlá. Tvůrce aplikace hned v popisu uvádí, že se v databázi nachází přesně 100 klasických českých receptů. Aplikace sice láká k použití mnoha dalšími funkcemi, jako je například vlastní nákupní seznam nebo možnost zvětšení písma přímo v nastavení aplikace, uživatelé jsou však velmi zklamaní, a to převážně z výše zmíněné málo obsáhlé databáze.

Při srovnávání s implementovanou aplikací Kuchařka, které se samozřejmě týká nabízených funkcí aplikace, Česká kuchařka [3] nenabízí žádné zajímavé funkce, které by neobsahovala i Kuchařka nebo jakákoliv jiná neplacená aplikace pro správu kuchařských receptů.

## 2.3 Miluji Vaření – recepty



Miluji Vaření – recepty [4] je další aplikace, která je použitelná pouze v online režimu. Podle oficiálního popisu v Obchodu Play je v aplikaci velmi zajímavá možnost uložení surovin, na které má uživatel aplikace alergii, do speciálního souboru a následně vyhledávané recepty budou filtrovány tak, aby tyto suroviny neobsahovaly. V receptech jsou také uvedeny nutriční a kalorické údaje, které však podle uživatelů aplikace nebývají přesné.

Tato aplikace na první pohled splňuje vše, co si běžný uživatel může přát, avšak podle komentářů v Obchodu Play v aplikaci nejvíce chybí možnost označení receptu jako oblíbený a následné filtrování receptů podle tohoto atributu.

## 2.4 Vaření.cz - Nejlepší recepty



Aplikace Vaření.cz [5] obsahuje podle oficiálního popisu v Obchodu Play nejlepší recepty z internetové stránky [www.vareni.cz](http://www.vareni.cz). Tato aplikace by podle popisku měla fungovat offline, ovšem podle komentářů k aplikaci je velká spousta receptů tvořena pouze odkazem na uvedené webové stránky, což uživatelům v offline režimu neposkytuje možnost uvařit jídlo podle daného receptu. Uživatelé aplikace si také stěžují na nemožnost přidání vlastního receptu.

Při srovnání aplikace Vaření.cz a Kuchařky je nejvíce bijící do očí možnost a nemožnost přidání vlastních receptů, jakož i úprava existujících receptů. Dalším nedostatkem této aplikace je oproti Kuchařce také nemožnost vyhledávání receptů pomocí suroviny nebo vytvoření vlastního nákupního seznamu.



Přehledná tabulka pro orientaci, co která aplikace obsahuje nebo naopak neobsahuje.

Název aplikace / název funkcionality	sRecepty	Česká kuchařka	Miluji Vaření - recepty	Vaření.cz	Implementovaná aplikace Receptář
Vytvoření nového receptu	NE	NE	ANO	ANO	ANO
Nákupní seznam	ANO	ANO	NE	NE	ANO
Offline režim	NE	ANO	NE	částečně	ANO
Časovač	NE	NE	NE	NE	ANO
Vyhledávání pomocí suroviny	NE	ANO	NE	NE	ANO
Sdílení receptů	ANO	ANO	NE	NE	ANO
Hodnocení receptů	ANO	NE	ANO	NE	ANO
Komentáře k receptu	ANO	ANO	ANO	NE	ANO
Sekce ‚Oblíbené recepty‘	NE	ANO	NE	NE	ANO
Přepočet počtu porcí v receptu	NE	NE	ANO	NE	ANO
Vyfocení a nahrání fotky k receptu	ANO	NE	ANO	NE	ANO
Různé jazykové verze	ANO	NE	NE	NE	NE
Videorecepty	ANO	NE	ANO	ANO	NE
Aplikace zdarma	ANO	NE	ANO	ANO	ANO

Tabulka 1: Srovnání existujících aplikací pro správu kuchařských receptů

Přínosem implementované aplikace Kuchařka oproti výše zmíněným aplikacím je především sloučení funkcionalit z těchto zmíněných a i z jiných, v některých případech dokonce i placených, aplikací. Například v žádné ze zmíněných aplikací nebyla nalezena (mimo jiné) možnost spuštění časovače výrobního procesu, natož pak možnost spuštění několika takovýchto časovačů.

Jak vyplývá z přehledu v tabulce, některé aplikace neumožňují ani tak základní funkcionalitu, jako je možnost vložení vlastního receptu do databáze nebo případně úpravu stávajících receptů. Tyto možnosti však převážně nejsou implementovány ze zcela logických důvodů, kterými je to, že aplikace, lépe řečeno databáze receptů, běží na online serveru.

Při výběru vhodné aplikace pro správu kuchařských receptů však samozřejmě vždy záleží na konkrétních požadavcích, které má daný uživatel na aplikaci.

### 3 Metody modifikace receptů

Modifikací receptů je především myšleno přepočítání množství surovin podle počtu porcí a případně také přepočet celkového času přípravy receptu i jeho následné tepelné úpravy v souvislosti s množstvím surovin.

Při průzkumu existujících možností těchto modifikací receptů bylo zjištěno následující.

Jakékoliv koeficienty pro přepočty množství surovin v receptech, pokud vůbec existují, nejsou využívány. Ve všech případech modifikací receptů, které byly prostudovány, byl využíván jednoduchý způsob kalkulace, kdy bylo zadáno pouze originální množství dané suroviny v receptu, originální počet porcí a následně nový počet porcí. Jednoduchým algoritmem poté došlo k výpočtu nového množství dané suroviny. Tento používaný algoritmus pro výpočet nového množství dané suroviny byl u všech případů následující.

$$(\text{originální\_množství} / \text{originální\_počet\_porcí}) * \text{nový\_počet\_porcí} = \text{nové\_množství}$$

*Rovnice 1: Algoritmus pro výpočet množství surovin*

Následně byl výsledek u některých zkoumaných zdrojů zaokrouhlen dolů na celé číslo. V jiných případech tvůrci receptu nezadali použité množství u surovin, kterých by se mohl týkat jiný koeficient používaný pro přepočet množství surovin než ten obvyklý. Příkladem takovýchto surovin je sůl nebo olej. A v neposlední řadě bylo také nalezeno velké množství aplikací a webových stránek s kuchařskými recepty, kde modifikovat recepty podle množství surovin vůbec nebylo možné.

Jelikož z průzkumu nevyplynula žádná jiná možnost, je v implementované aplikaci Receptář využito také pouze jednoduchého, výše zmíněného, algoritmu, pro přepočet množství surovin podle zadaného počtu porcí.

V průběhu zkoumání metod modifikací receptů nebyla nalezena ani jakákoliv informace týkající se úpravy časového intervalu pro přípravu ani tepelnou úpravu receptu. Z tohoto vyplývá jediné, a tedy, že ani v implementované aplikaci Receptář není využita žádná úprava či přepočet času, který uživatel stráví při pracovním postupu výroby receptu pro různé počty porcí.

## 4 Analýza požadavků kladených na aplikaci

Při vytváření modelu projektu byly specifikované základní vlastnosti aplikace a její funkcionality, která byla (v některých případech) v průběhu implementace rozšiřována. Pro vyjádření celkové specifikace funkcionalit aplikace bylo využito Use Case Diagramu [7][8], který umožňuje zobrazit chování aplikace tak, jak ho vidí uživatel. Use Case Diagramy obecně slouží převážně k popisu funkcionality systému, ovšem již dále nepopisují, jak je toho dosaženo. K dalšímu využití Use Case Diagramů patří také usnadnění komunikace mezi zákazníkem nebo uživatelem aplikace a týmu vývojářů při vyjasňování cílů aplikace.

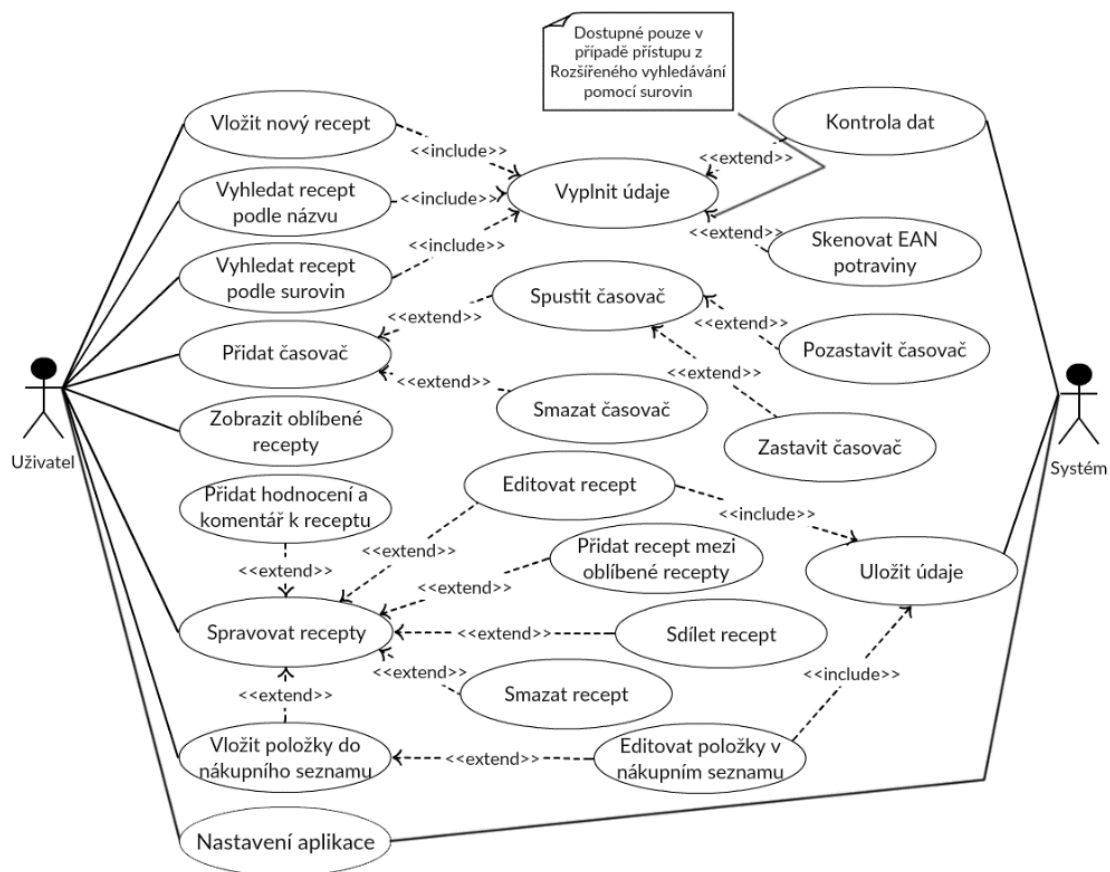
Pro popsání struktury a funkcionality klíčového prvku aplikace, kterým je možnost vyhledávání receptů pomocí surovin, je dále využito Třídního a Sekvenčního diagramu. Diagramy tříd [10] patří do skupiny diagramů struktur, a tedy nabízí vynikající možnost vizualizace struktury systému. Třídní diagramy zobrazují statický pohled na systém, tj. zejména třídy jako typy objektů, obsah tříd a statické vztahy, které mezi nimi existují. Sekvenční diagramy [9] patří do skupiny diagramů interakce, které ukazují, jak komunikují jednotlivé objekty mezi sebou, a to v časové rovině. Sekvenční diagramy podávají jasný obraz o činnosti několika účastníků během modelované interakce daného případu užití.

Následně je zde také popsána struktura databáze receptů a EANů pomocí relačního datového modelu [18]. Tento model sdružuje data do tzv. relací (tabulek), které obsahují n-tice (řádky). Tabulka je struktura záznamů s pevně stanovenými položkami. Každý sloupec má definován jednoznačný název a typ. V těchto tabulkách jsou prováděny veškeré databázové operace.

### 4.1 Use Case diagram

Na obrázku 1, který znázorňuje diagram případu užití, jsou přehledně uvedeny veškeré možnosti, které jsou dostupné pro všechny uživatele aplikace Receptář.

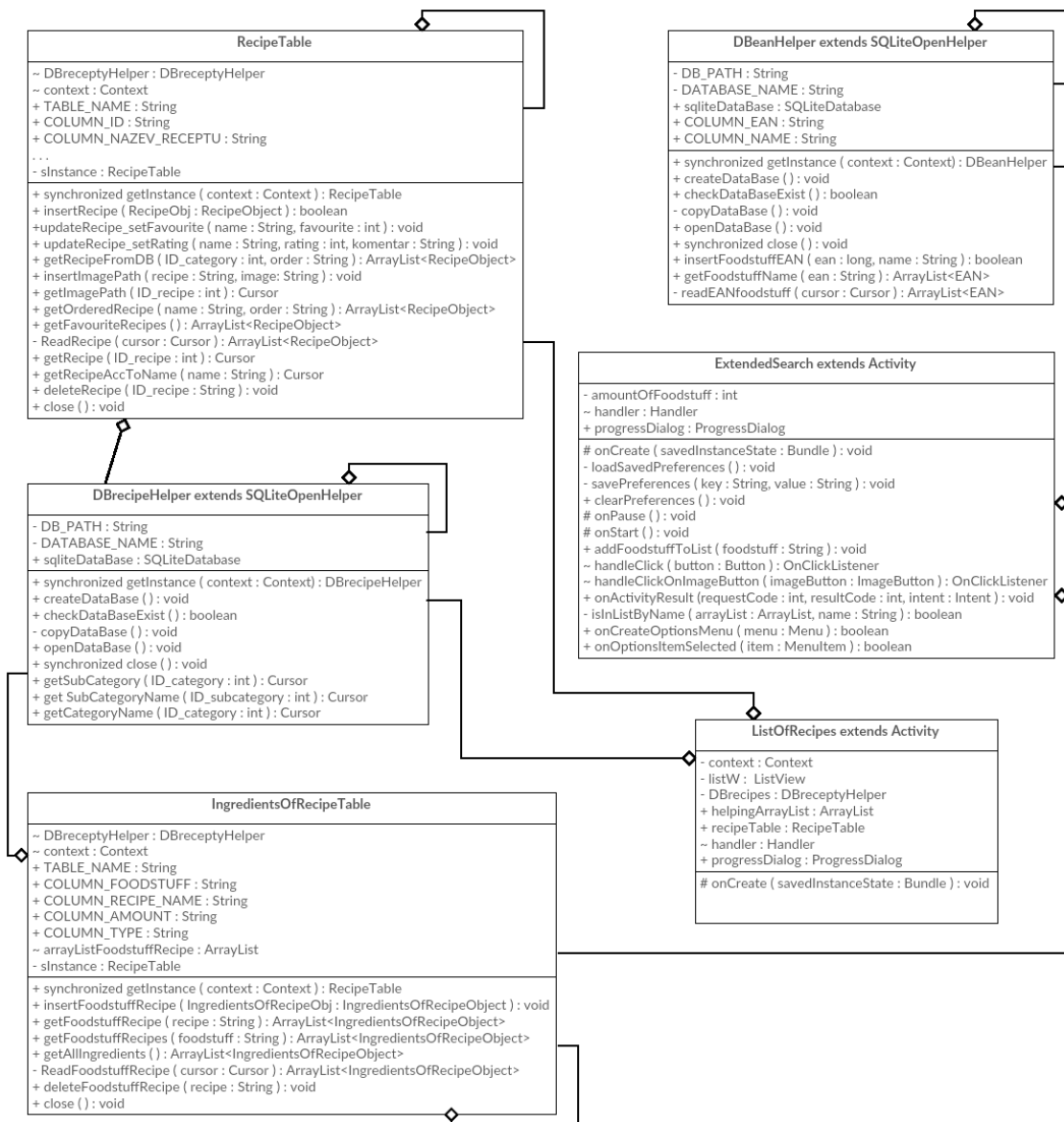
Patří mezi ně mimo jiné například Správa receptů, do které spadají možnosti, jako je úprava nebo sdílení daného receptu. Důležitou součástí aplikace je také možnost přidání a následné spuštění časovače výrobního procesu a možnost vyhledávání receptů ne pouze pomocí názvu receptu, ale také pomocí surovin.



Obrázek 1: Use Case diagram funkcionalit aplikace

## 4.2 Třídní diagram

Klíčový prvek aplikace, kterým je možnost vyhledávání receptů pomocí surovin, a to ať již pomocí naskenování EANu potraviny anebo samotným zadáním názvu potraviny, je detailněji popsán pomocí níže zobrazeného Třídního diagramu (Obrázek 2).

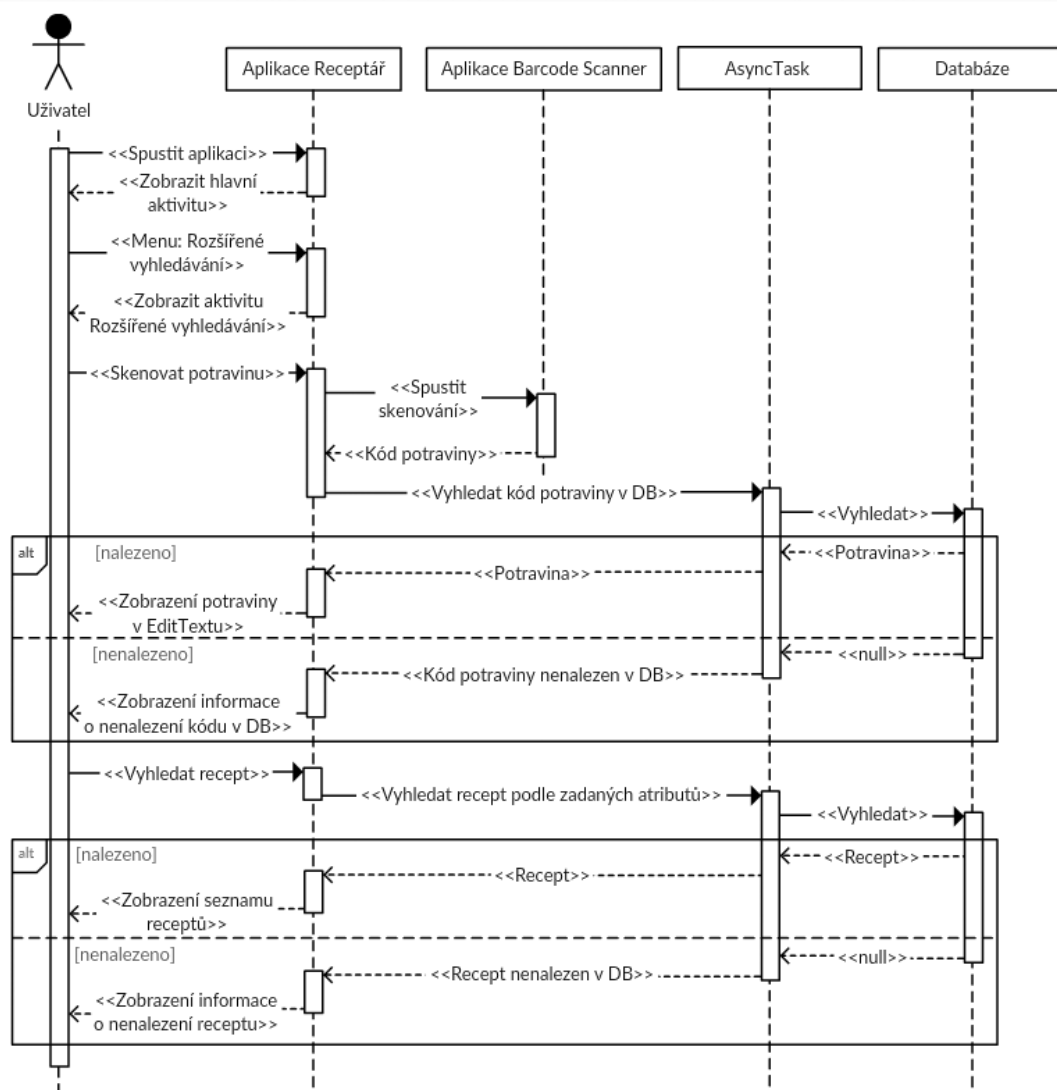


Obrázek 2: Třídní diagram, rozšířené vyhledávání

## 4.3 Sekvenční diagram

Další diagram popisující funkcionalitu klíčového prvku aplikace je diagram sekvenční (Obrázek 3).





Obrázek 3: Sekvenční diagram, vyhledávání receptu pomocí naskenování EANu suroviny

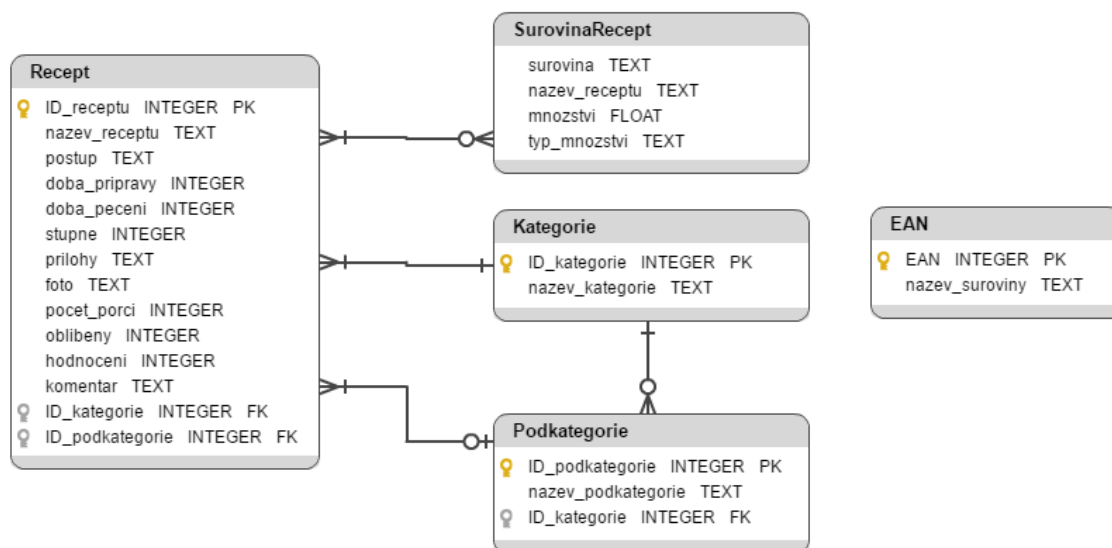
Jak lze vidět na tomto diagramu, funkce Skenování potraviny spouští druhou aplikaci s názvem Barcode Scanner, která provádí rozpoznání obrazu a samostatné oskenování EANu potraviny. Následně tato aplikace vrací naskenovaný kód potraviny jako textový řetězec, složený z číslic. Tento kód je následně použit pro vyhledání dané potraviny v lokální databázi čárových kódů potravin. V této databázi je provedeno vyhledání potraviny podle kódu a následné zaslání názvu nalezené potraviny zpět do zobrazovacího vlákna, kde je název potraviny zobrazen v editovatelném textovém poli.

Samotné vyhledávání receptů poté funguje pomocí zřetězení názvů všech zadaných potravin a do databáze je zaslán dotaz na filtraci receptů podle surovin, které jsou do daných receptů přiřazeny. V případě, že žádný recept se zadanými surovinami neexistuje, je uživateli vypsána tato informace do místa, kde by byl v opačném případě zobrazen seznam receptů, obsahující dané suroviny.

## 4.4 Návrh databáze – relační datový model

V aplikaci jsou použité celkem dvě vzájemně nezávislé databáze. Jedna obsahuje veškeré informace související s recepty, ať už se jedná o suroviny, kategorie receptů nebo výrobní postup. Druhá databáze obsahuje informace související s čárovými kódy potravin.

Pro pochopení databáze a jejího obsahu je zde uveden relační datový model obou databází použitých v aplikaci.



Obrázek 4: Relační databázový model - Recepty.db, EANy.db

### 4.4.1 Kardinalita vztahů entit

Kardinalita jednotlivých vztahů mezi tabulkami databáze je vcelku jasná již z výše zmíněného ER diagramu, ale pro úplnost je zde uvedena přehledná tabulka.

<b>Recept – SurovinaRecept</b>	M:N
<b>Recept – Kategorie</b>	N:1
<b>Recept – Podkategorie</b>	N:1
<b>Kategorie – Podkategorie</b>	1:N

Tabulka 2: Kardinalita vztahů entit v databázi

## 5 Uživatelské rozhraní

Jelikož operační systém Android běží na velké spoustě různě velkých zařízeních, bylo nutné navrhnout takové GUI, které bude jednoduché a intuitivní na všech velikostech zařízení a tedy i výsledné zobrazení receptu bude vypadat přibližně stejně.

Aplikace byla vyvíjena a testována na obrazovce s rozměry 4,5“ a verzí Androidu 5.1.

### 5.1 Hlavní obrazovka

Uživatel po zapnutí aplikace okamžitě vidí vyhledávací pole pro rychlé nalezení receptu podle jeho názvu. Takto je GUI navrženo z jednoduchého důvodu. Z vlastní zkušenosti vím, že nejčastěji uživatel hledá daný recept právě podle jeho názvu.

Pokud by si však uživatel chtěl jen tak procházet kategoriemi a doufat, že nalezne inspiraci v seznamu receptů, je pro něj nejrychlejší volbou vybrat si jeden ze šesti obrázků. Ty jsou umístěny hned pod vyhledávacím poličkem. Tyto obrázky výstižně zastupují kategorie receptů, které jsou členěny následovně: snídaně, oběd, večeře, svačina, zákusky a nápoje. Obrázková verze menu je zvolena pro intuitivnější vybírání kategorie.



Obrázek 5: Vzhled hlavní obrazovky

V hlavním menu Receptáře pak uživatel nalezne veškeré hlavní funkce aplikace. Ty jsou následující.

- Rozšířené vyhledávání pomocí surovin
- Zobrazit oblíbené recepty
- Vytvořit nový recept
- Nákupní seznam
- Časovač

Tyto části aplikace a jejich UI jsou popsány níže.

## 5.2 Zobrazení detailu receptu

Uživatelské rozhraní při zobrazování konkrétního receptu je řešeno pomocí aktuálně velmi oblíbených posuvných karet (tzv. *SlidingTabs*). Tyto karty fungují na principu posouvání obsahu zprava doleva a naopak. Uživatel má také možnost jednoduše přepínat mezi kartami pouhým klikem na název dané karty.

V detailu receptu je implementována také možnost označení receptu jako oblíbený recept pomocí zaškrťovacího tlačítka. Takto označené recepty si následně může uživatel zobrazit při zvolení položky „Zobrazit oblíbené recepty“ z hlavního menu aplikace. Do detailu receptu je také logicky zařazena fotografie, kterou může uživatel přiřadit k receptu výběrem z galerie fotografií anebo vyfotografováním nového obrázku. Tuto fotografii je také možno zvětšit na celou obrazovku, a to pomocí kliknutí na obrázek (tato možnost však není funkční, pokud k receptu není přiřazen žádný obrázek).

### 5.2.1 Karta Přehled

Uživatel aplikace ze všeho nejdřív vidí kartu „Přehled“, kde se mimo jiné dozví hodnocení receptu nebo dobu přípravy receptu a následnou dobu pečení nebo vaření.

V této kartě je také možnost zapnutí časovače výrobního procesu, který je implementován tak, aby se uživateli po stisknutí ikonky časovače zobrazila možnost přidání nového časovače do seznamu časovačů, následně uživateli stačí kliknout na ikonku spuštění časovače.

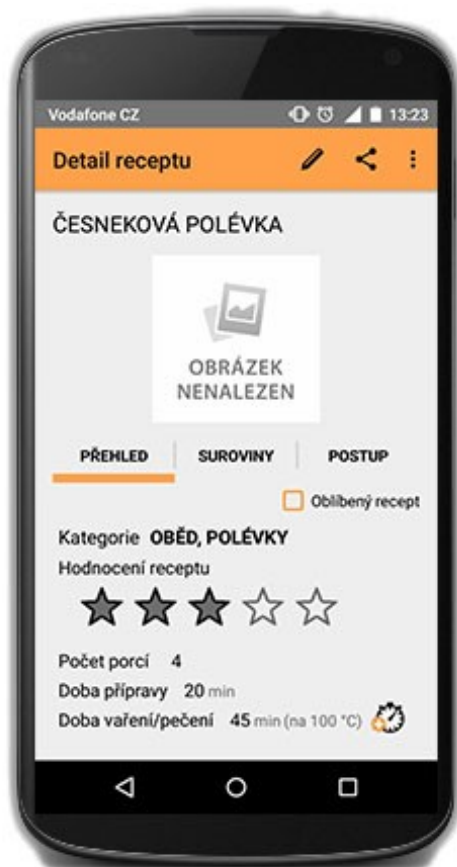
### 5.2.2 Karta Suroviny

Následně recept obsahuje kartu Suroviny. Jak již z názvu vyplývá, zde se uživatel dozví, z jakých surovin a z jakého množství se daný recept skládá.

Zde má uživatel aplikace také možnost modifikovat množství surovin, a to podle počtu porcí. Také je zde možnost přidávat tyto suroviny do implementovaného nákupního seznamu, a to pouze klikem na ikonu přidání do nákupního seznamu.

### 5.2.3 Karta Postup

Další a zároveň poslední posuvnou kartou je karta Postup. Zde uživatel nalezne to nejdůležitější z celého receptu – pracovní postup a vhodné přílohy k receptu.



Obrázek 6: Zobrazení detailu receptu

V menu zobrazeného receptu uživatel dále nalezne různé možnosti jako je například úprava receptu (ikona tužky), sdílení (ikona pro sdílení), ohodnocení a okomentování receptu nebo také smazání celého receptu. V možnostech je také náhled na nákupní seznam a možnost přidání nebo změnění obrázku. Tyto části aplikace a jejich UI jsou popsány níže.

## 5.3 Úprava receptu

Uživatel může provádět úpravy kteréhokoliv receptu, ovšem aby toto mohl provést, musí si z menu v zobrazeném receptu zvolit možnost „Upravit recept“ (ikona tužky). Poté už může například měnit množství surovin nebo modifikovat samotný postup přípravy receptu. Tyto změny v receptu se ovšem projeví jen v lokální databázi. Pokud by se uživatel chtěl o upravený recept podělit se svým okolím, poskytuje mu aplikace možnost sdílení receptu mimo jiné pomocí G-mailu, Facebooku, Hangouts nebo i pomocí SMS zprávy.

## 5.4 Hodnocení receptu

Ohodnotit nebo okomentovat recept je pro uživatele Receptáře velmi snadné. Ačkoli komentář i hodnocení je dostupné pouze pro toho jednoho určitého uživatele, je možné toto hodnocení sdílet s ostatními, a to stejným způsobem, jako se sdílí samotný recept.

Pokud má recept hodnocení a (nepovinný) komentář, je tato informace automaticky připojena k receptu a tedy pokud se uživatel rozhodne sdílet tento recept, sdílí s ním i své hodnocení.

Tuto možnost aplikace poskytuje nejen z důvodu možnosti jednoduchého a intuitivního hodnocení receptu pomocí hvězdiček, ale také k přidání komentáře k tomu, jak daný recept chutnal přátelům nebo k připsání podrobnějšího postupu v případě, že si uživatel chce ponechat i předchozí pracovní postup. Komentář k receptu může tedy sloužit i jako poznámkový blok.

V závislosti na hodnocení receptu aplikace nabízí také možnost seřazení receptů podle uživatelského hodnocení, podobně jako filtrace oblíbených receptů. Toto řazení receptů podle hodnocení si uživatel může nastavit v nastavení aplikace, dostupné z hlavní stránky.

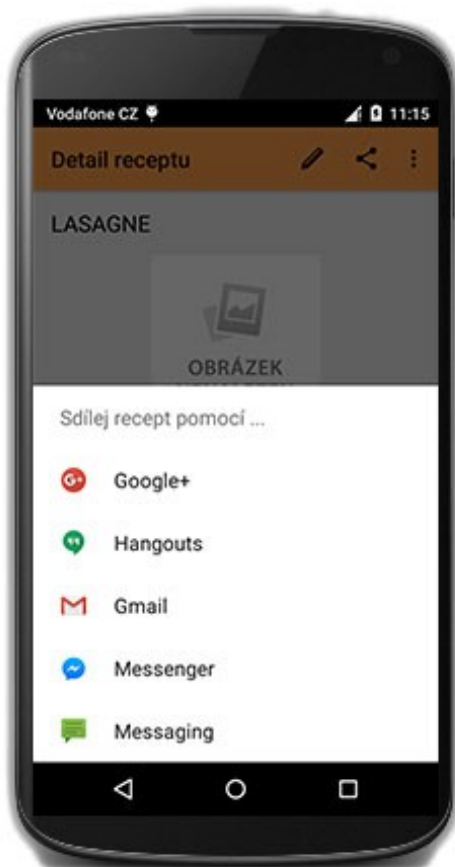
## 5.5 Vytvoření nového receptu a sdílení receptů

Aplikace umožňuje také vytvořit zcela nový recept, který je následně uložen do lokální databáze. Při tvorbě nového receptu je však nutné dodržet podmínku, že v každém políčku musí být něco uvedeno. Také není možné uložit dva recepty pod stejným názvem. Pokud uživatel má více verzí daného receptu, je logicky stejně lepší, pojmenovat si tyto verze zvlášť (například „Rizoto - od babičky“) pro lepší orientaci.

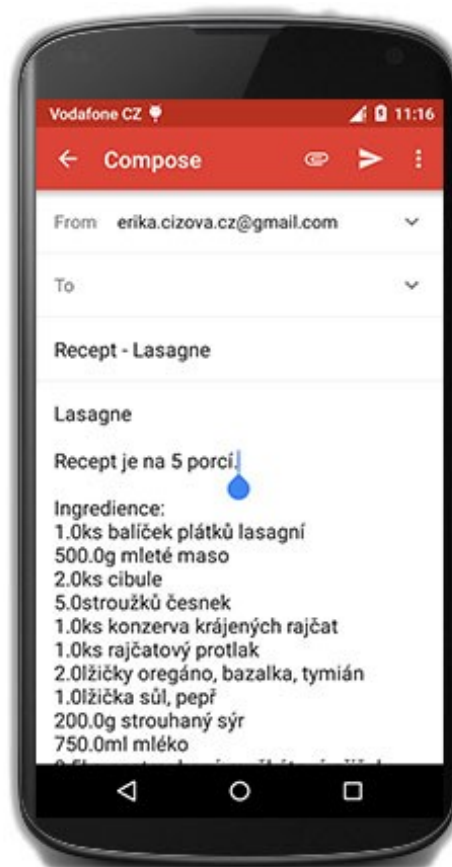
Pokud se uživatel následně chce o svůj nový recept nebo další verzi receptu podělit se svým okolím, poskytuje mu aplikace několik možností sdílení receptu, například pomocí G-mailu nebo Facebooku.

Pro tuto možnost je samozřejmě zapotřebí připojení k Internetu (ačkoliv ve většině případů lze „rozepsanou zprávu“ uložit pro odeslání jakmile bude k dispozici síť Wi-Fi) a vytvořený účet v dané aplikaci.





Obrázek 7: Možnosti sdílení receptu



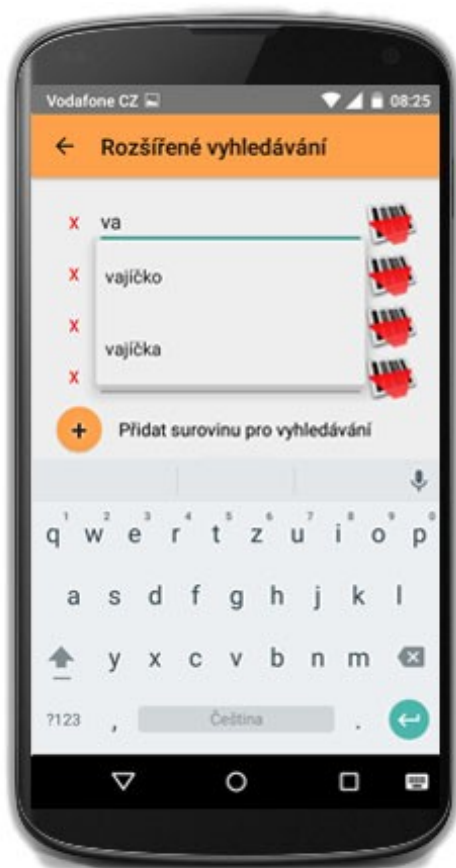
Obrázek 8: Sdílení pomocí G-mailu

## 5.6 Vyhledávání receptu podle názvu i podle surovin

Uživatel má dvě možnosti, jak vyhledávat recepty. Jeden je klasický, kdy uživatel zadá název receptu. Druhá možnost se uživateli naskýtá při výběru položky „Rozšířené vyhledávání“ z hlavního menu aplikace.

Tato možnost je skvělou volbou pro takové uživatele, kteří hledají inspiraci anebo by rádi vyzkoušeli nové recepty. Uživatel jednoduše zadá, ze kterých surovin chce vařit a aplikace mu nabídne vhodné recepty. Ale to není všechno. Suroviny uživatel nemusí zadávat jen ručně, má také možnost vyfotografovat EAN dané suroviny.

Pro tuto možnost však aplikace vyžaduje nainstalování další, jednoduché aplikace, Barcode Scanner [6], která pomocí kamery telefonu rychle oskenuje a rozezná čárové kódy potravin. Tato aplikace je vybavena automatickým ostřením a rychlým rozpoznáním obrazu. Stará se také o dekodování obrazu a dále už jen posílá získaná data zpět do Receptáře, kde se o rozeznání, o jakou potravinu se jedná, stará další lokální databáze.



Obrázek 9: Rozšířené vyhledávání, zadání suroviny



Obrázek 10: Rozšířené vyhledávání, skenování EANů

## 5.7 Spravování databáze EANů

Pod rozšířené vyhledávání pomocí surovin se zcela intuitivně řadí i možnost upravovat databázi čárových kódů. Uživatel může pomocí vyfotografování čárového kódu potravin přidat tuto potravinu do databáze. V případě, že daná potravina již v databázi existuje, má uživatel možnost tuto položku databáze upravit a následně znovu uložit.

Databáze obsahuje jen určitý základní set (zhruba 50) čárových kódů a je jen na uživateli aplikace, jestli jej bude rozšiřovat, nebo nikoli. Jelikož však existuje velké množství potravin a jejich značek, není snad ani možné mít databázi čárových kódů opravdu kompletní.

V průběhu implementace Receptáře došlo samozřejmě k průzkumu již existujících databází EANů, ale bohužel veškeré obsáhlejší databáze jsou v angličtině a většinou se na ně vztahují určitá autorská práva a pro jejich použití by bylo nutné zaplacení ne zrovna malé finanční částky. Z tohoto důvodu tedy databáze lokální, obsahující pouze lokální potraviny od místních dodavatelů.

## 5.8 Časování výrobního procesu – Časovač

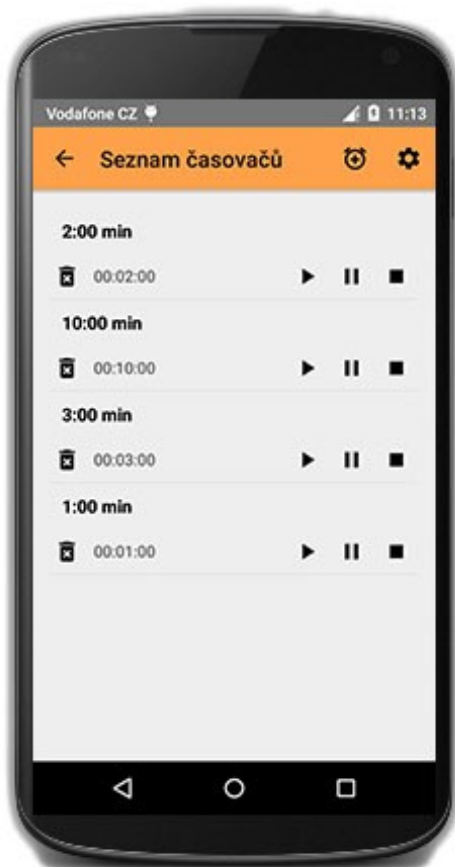
Možnost nastavení časovačů implementovaných do aplikace pomáhá uživateli časovat výrobní proces. Aktivita časovače se skládá ze zobrazení existujících časovačů v seznamu a samozřejmě také možnosti přidat další, nový, časovač. V seznamu má uživatel možnost kterýkoliv časovač spustit (a může jich také spustit více najednou), pozastavit nebo zastavit a samozřejmě také smazat. Jelikož tato část aplikace po spuštění časovače zůstává běžet i po zavření, upozorní uživatele (ve výchozím nastavení) vibrováním a výchozím tónem vyzvánění na to, že stanovená doba vypršela. To pro uživatele znamená jediné: jeho recept by měl být hotov.

Upozornění v základním nastavení není nutné vypínat, protože samo trvá jen několik málo sekund, a poté se samostatně ukončí. Uživatel aplikace tedy nemusí například rychle umývat ruce jen proto, aby vypnul upozornění na zařízení.

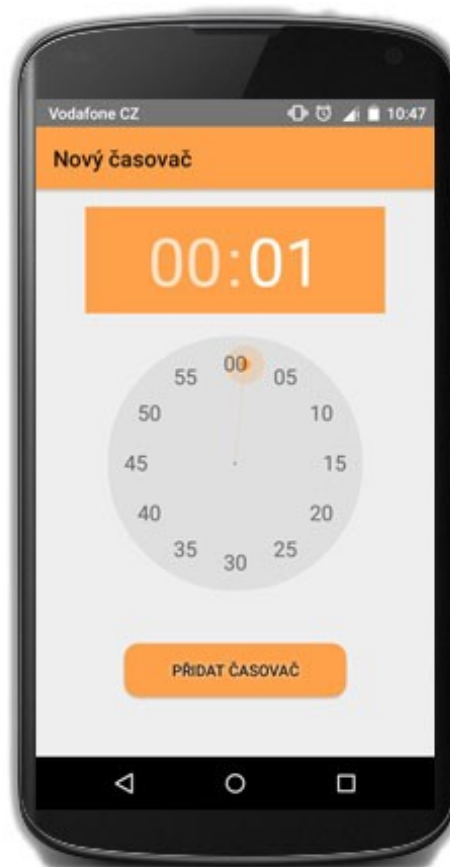
Toto nastavení však skýtá jeden problém. Co když uživatel přeslechne upozornění na zařízení, protože zrovna musel odejít z kuchyně, z dosahu zařízení?

Z tohoto důvodu je implementováno nastavení upozornění, kdy si uživatel může vybrat ze dvou možností. Jedna je výše popsaná, tedy že telefon uživatele jednou upozorní a sám upozorňování po několika sekundách zastaví. Druhá možnost je logicky taková, že si uživatel nastaví, aby upozornění v zařízení pokračovalo po tak dlouhou dobu, dokud jej sám nevypne. Mimo to si uživatel může také nastavit vlastní tón pro upozornění a také může vypnout či zapnout vibrování při tomto upozornění.

Nabídku nastavení časovače lze vyvolat jak z hlavního menu aplikace, tak z menu Časovače, pro snadnější přístup k nastavení.



Obrázek 11: Seznam časovačů



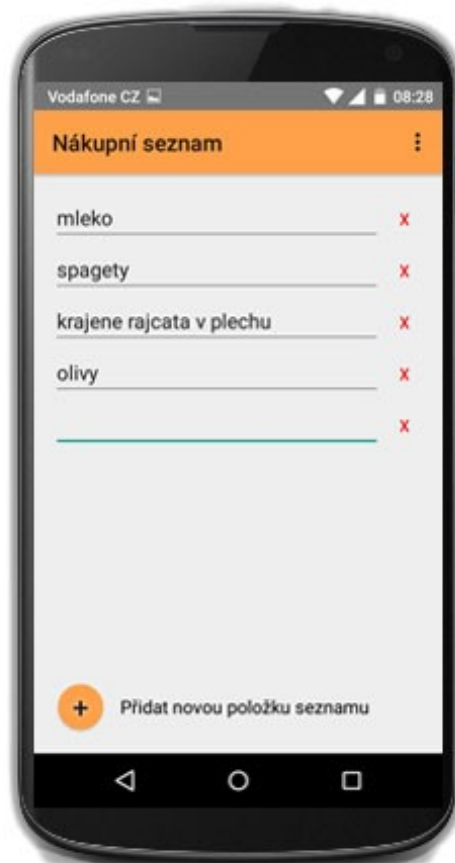
Obrázek 12: Vytvoření nového časovače

## 5.9 Nákupní seznam

Implementace vlastního nákupního seznamu umožňuje uživateli importovat suroviny přímo ze zobrazeného receptu do seznamu pouhým stiskem ikony pro přidání suroviny do nákupního seznamu. Další možností, jak přidat nebo upravit položky seznamu, je samozřejmě vyvoláním nabídky menu a vybrání možnosti ‚Nákupní seznam‘, který je přístupný jak z hlavního menu aplikace, tak z menu v detailu receptu.

Zde uživatel nalezne zcela intuitivně rozmístěné GUI, kdy si pomocí tlačítka přidává řádky neboli položky v seznamu. Každou takto přidanou položku seznamu může pomocí malého červeného tlačítka ‚x‘ odebrat, takže je snadné odškrtnout položku po položce podle toho, co již je v nákupním vozíku.

Uživatel po přidání nové položky do seznamu (nebo její úpravy, případně smazání) nemusí hlídat, jestli má aktuální verzi nákupního seznamu uloženou. Ukládání nákupního seznamu je totiž řešeno automaticky – při opuštění této aktivity aplikace samostatně uloží aktuální verzi do paměti telefonu.



Obrázek 13: Nákupní seznam

V menu nákupního listu je zavedena možnost vymazání celého vytvořeného seznamu, a to z důvodu snazší administrace.

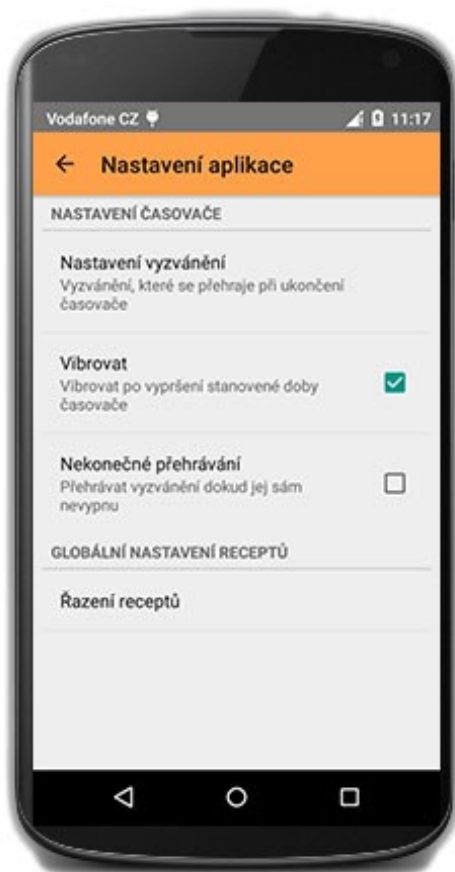
## 5.10 Zobrazení oblíbených receptů

Při zobrazování oblíbených receptů z Receptáře je využíváno jednoduchého algoritmu, který z databáze receptů vyfiltruje jen ty recepty, které mají příznak ‚oblíbený‘. Pro uživatele je takto o mnoho snadnější nalézt vyzkoušený recept, pokud ví, že si jej označil jako oblíbený, ale už například neví, jak se přesně jmenoval. UI této funkcionality aplikace lze vidět na obrázku 15.

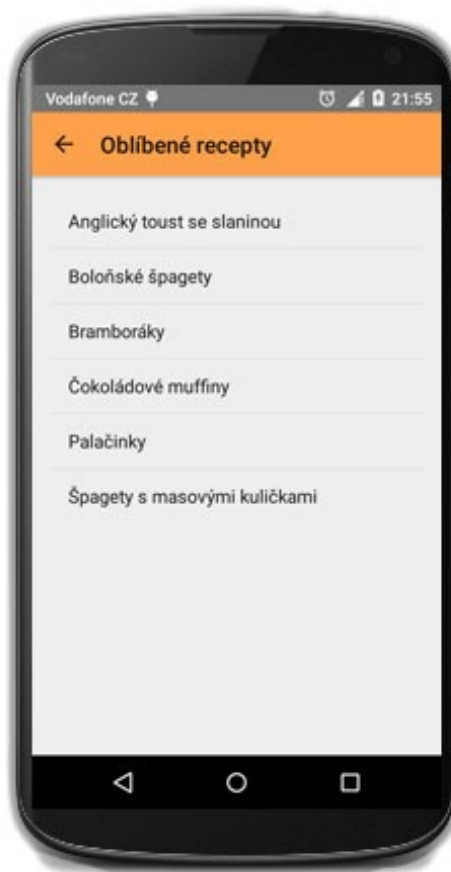
## 5.11 Nastavení aplikace

Aplikace umožňuje určité možnosti nastavení, z nichž se velká část týká nastavení časovače. Uživatel si tak může například vybrat, jaký tón upozornění bude přehrán při dokončení odpočtu času, nebo zda bude telefon vibrovat. Jednou možností z těchto nastavení je také možnost nastavit k upozornění tzv. nekonečné vyzvánění, kdy aplikace bude uživatele tak dlouho upozorňovat na vypršenou stanovenou dobu, dokud toto upozornění uživatel sám nevypne.

Dále má uživatel možnost vybrat si, jakým způsobem budou seřazeny vyhledávané recepty. Výchozí nastavení je řazení podle abecedy, avšak uživatel může také zvolit možnost řazení podle jeho vlastního, uloženého, hodnocení receptů. Třetí možností seřazení receptů je pomocí doby přípravy receptu a jeho následné tepelné úpravy. Tyto dvě hodnoty jsou před seřazením receptů sečteny a pomocí takto vzniklé doby jsou recepty seřazeny od nejkratší celkové doby přípravy po nejdelší. Uživateli tedy vzniká filtrace receptů podle doby, kterou stráví přípravou a tepelnou úpravou receptu.



Obrázek 14: Nastavení aplikace



Obrázek 15: Zobrazení oblíbených receptů



## 6 Implementace Receptáře

### 6.1 SQLite databáze

Pro přístup k SQLite databázi receptů je využíváno návrhového vzoru Jedináček neboli Singleton [11]. Tento návrhový vzor se stará o to, aby třída, která je využívána pro přístup k databázi, měla vždy pouze jednu instanci a aby byl k této instanci umožněn globální přístup.

Důvodem k použití tohoto návrhového vzoru bylo mimo jiné zrychlení aplikace díky používání jedné instance namísto vytváření vždy nového připojení k databázi a jeho následné ukončování. Dalším důvodem byl častý únik databáze (*database leak*), který byl způsoben nedokonalým přístupem k databázi. Tento problém byl spolu s implementací návrhového vzoru vyřešen.

#### 6.1.1 Využití asynchronního přístupu pro práci s databází

Při získávání dat z databáze a jejich následnému zobrazování v různých aktivitách aplikace je využito `AsyncTasku` [12], neboli asynchronního přístupu k databázi a následné aktualizaci UI aplikace. Je známo, že není doporučováno provádět dlouhé procesy na UI vláknech, jelikož toto UI aplikace „zamrzne“ dokud není daný proces dokončen. Z tohoto důvodu poskytuje `AsyncTask` řádné a hlavně velmi snadné použití tohoto UI vlákna. Tato třída totiž poskytuje možnost provádět dlouhé operace na pozadí a výsledky operací jsou poté automaticky zveřejněny na UI vláknech bez nutnosti manipulace s tímto vláknem nebo s manipulátory (*handlers*) vláken. Typicky se jedná právě o práce s databází nebo stahování souborů z internetu.

`AsyncTask` je navržen jako tzv. pomocník neboli pomocná třída, která usnadňuje komunikaci mezi IU a tzv. *worker* vláknem neboli vláknem běžícím na pozadí aplikace. Toto tzv. *worker* vlákno provádí procesy, které nejsou okamžité, tedy takové, které trvají delší dobu. Nicméně je potřeba si uvědomit, že UI vlákno není možné aktualizovat z kteréhokoliv jiného vlákna. Aby bylo možno upravovat UI vlákno, poskytuje Android několik možností, z nichž jedna je právě `AsyncTask`. Další možnosti jsou použití metod `Activity.runOnUiThread(Runnable)`, `View.post(Runnable)` nebo `View.postDelayed(Runnable, long)`.

Pro využití funkcionality `AsyncTasku` a jeho možnosti přístupu k UI vláknem, je nutné vytvořit podtřidu `AsyncTasku`. Následně je nutné implementovat metody `doInBackground(Params...)` a `onPostExecute(Result)`. Tyto dvě metody zajišťují správnou funkcionality třídy `AsyncTask`. Metoda `doInBackground()` běží v tzv. *poolu*, kde jsou umístěny i další vlákna, běžící na pozadí aplikace. Tato metoda v aplikaci Receptář zajišťuje přístup k databázi a získávání dat z databáze. Abychom mohli aktualizovat UI podle dat, která získáme z databáze, je nutné použít metodu `onPostExecute()`, která posílá výsledky získané z metody `doInBackground()` do vlákna, které se stará o UI. Díky této metodě je umožněna bezpečná aktualizace UI ve spuštěné aktivitě. Pro spuštění asynchronního přístupu k datům v databázi je nutné pouze volat metodu `execute()` z běžícího UI vlákna aplikace.

Při využívání podtřídy `AsyncTasku`, je také možno implementovat další metody, jako jsou `onPreExecute()` a `onProgressUpdate()`. Metoda `onPreExecute()` je volána před tím, než je spuštěno samotné vlákno na pozadí. Nejčastěji je tohoto využito pro nastavení úlohy, která bude běžet na pozadí (například zobrazení ukazatele průběhu). Metoda `onProgressUpdate()` je volána, pokud je využito metody `publishProgress()` v metodě `doInBackground()` a je využívána pro zjištění progresu vlákna a zobrazení tohoto progresu v UI vlákně.

## 6.2 Použití aplikace Barcode Scanner

Při rozšířeném vyhledávání receptů pomocí zadání suroviny pomocí oskenování EANu suroviny je využito další aplikace – Barcode Scanner [6]. Tato aplikace je zavolána stiskem ikonky skenování, která pomocí kamery telefonu rychle oskenuje a rozezná čárový kód potravin.

Tato aplikace je vybavena automatickým ostřením a rychlým rozpoznáním obrazu a také se stará o dekodování obrazu. Dále už jen posílá získané data zpět do Receptáře, kde se o rozeznání, o jakou potravinu se jedná, stará další lokální databáze.

## 6.3 Přidání obrázku do receptu a jeho následné zobrazení v detailu receptu

Ve výchozím obsahu databáze není u žádného z receptů uložena fotografie, avšak uživatel má možnost fotografii k receptu přiložit. Přiložení fotografie lze uskutečnit dvěma základními způsoby. Jedním z nich je vybrání fotografie z galerie telefonu a druhým způsobem je samozřejmě vyfotografování právě vytvořeného pokrmu podle receptu.

Pro vyvolání nabídky těchto dvou možností je nutné být v zobrazení detailu receptu a následně z menu vybrat možnost „Přidat obrázek“.

```
private void selectImage() {
    final CharSequence[] items = { "Vyfoť nový obrázek", "Vyber obrázek z
        galerie", "Zrušit" };
    AlertDialog.Builder builder = new AlertDialog.Builder(Recept.this);
    builder.setTitle("Přidat obrázek ...");
    builder.setItems(items, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int item) {
            if (items[item].equals("Vyfoť nový obrázek")) {
                Intent ziskatFoto = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                ziskatFoto.putExtra(MediaStore.EXTRA_OUTPUT, picUri);
                startActivityResult(ziskatFoto,
                    CAPTURE_IMAGE_ACTIVITY_REQUEST_CODE);
            } else if (items[item].equals("Vyber obrázek z galerie")) {
```

```

Intent intent = new Intent();
intent.setType("image/*");
intent.setAction(Intent.ACTION_GET_CONTENT);
startActivityForResult(Intent.createChooser(intent, "Select
    File"), IMAGE_FROM_FILE_ACTIVITY_REQUEST_CODE);
} else if (items[item].equals("Zrušit")) {
    dialog.dismiss();
}
}
});
builder.show();
}

```

*Zdrojový kód 1: Nabídka možností přidání obrázku k receptu*

### 6.3.1 Výběr obrázku z galerie

Pro výběr obrázku z galerie je využito možnosti zobrazení galerie zařízení, kde má uživatel možnost výběru obrázku z kteréhokoliv uloženého alba. Aplikace u takto vybraného obrázku zjistí cestu k obrázku a tu následně v metodě `onActivityResult()` uloží do databáze pro příští použití.

```

private void onSelectFromGalleryResult(Intent data) {
    if (data != null) {
        try {
            Uri selectedImageURI = data.getData();
            File destination = new File(Environment
                .getExternalStorageDirectory() + "/DCIM/Camera" +
                File.separator + getRealPathFromURI(selectedImageURI));
            picUri = Uri.fromFile(destination);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

*Zdrojový kód 2: Výběr obrázku z galerie*

### 6.3.2 Vyfotografování nového obrázku

Pro vyfotografování nového obrázku je využito existující aplikace Camera, která aplikaci v metodě `onActivityResult()` vrací cestu k uloženému obrázku. Takto získaná cesta je

následně uložena do databáze receptů, která je v případě načtení daného receptu použita pro nalezení a následné zobrazení obrázku.

```
private void onCaptureImageResult(Intent data) {
    { ... }

    String timeStamp = new SimpleDateFormat("yyyyMMdd_HH:mm:ss").format(new
Date());
    Bitmap thumbnail = (Bitmap) data.getExtras().get("data");
    ByteArrayOutputStream bytes = new ByteArrayOutputStream();
    thumbnail.compress(Bitmap.CompressFormat.JPEG, 50, bytes);
    File destination = new File(mediaStorageDir.getPath() +
        + File.separator + "IMG_" + timeStamp + ".jpg");
    { ... }
    picUri = Uri.fromFile(destination);
}
```

*Zdrojový kód 3: Vyfotografování nového obrázku*

### 6.3.3 Zobrazení obrázku v aplikaci

Jakmile je cesta k obrázku uložena do lokální databáze, je tato cesta dekodována na samotný obrázek a ten je následně při zobrazení detailu receptu zobrazen na určeném místě v aplikaci. Pokud se ovšem obrázek na uložené cestě nenachází (uživatel tento obrázek smazal), je v aplikaci zobrazen výchozí obrázek (,Obrázek nenalezen'). Tento výchozí obrázek uživatele pouze upozorňuje, že k danému receptu nebyl nalezen žádný obrázek. Stejný obrázek se uživateli zobrazí také v případě, že k receptu ještě zatím nebyl žádný obrázek přiložen.

```
final String imagePathString =
image.getString(image.getColumnIndex(DBr.COLUMN_FOTO));
final Bitmap smallerImageBitmap =
BitmapFactory.decodeFile(imagePathString, bmOptions);
if (smallerImageBitmap == null){
    foto.setImageResource(R.drawable.noimagefound);
}else {
    foto.setImageBitmap(smallerImageBitmap);
}
```

*Zdrojový kód 4: Zobrazení obrázku*

Jestliže je k receptu přiřazen obrázek jiný než výchozí (,Obrázek nenalezen'), pak se klikem na tento obrázek uživateli zobrazí daný obrázek ve větším formátu. Tohoto je docíleno pomocí nové aktivity, do které jsou posílány data o cestě k obrázku. V této nové aktivitě je obrázek následně zobrazen v originální velikosti.

```

public class FullScreenImage extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
                               WindowManager.LayoutParams.FLAG_FULLSCREEN);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity_full_screen_image);
        ImageView imageView = (ImageView) findViewById(R.id.imageView2);
        Bundle extras = getIntent().getExtras();
        if (extras != null) {
            String imagePath = extras.getString("image");
            final Bitmap imageBitmap = BitmapFactory.decodeFile(imagePath);
            imageView.setImageBitmap(imageBitmap);
        }
    }
}

```

*Zdrojový kód 5: Zobrazení obrázku na celou obrazovku*

Ať už si však uživatel vybere jakoukoli možnost vložení obrázku k receptu, obrázek není možno dále sdílet s ostatními pomocí sdílení receptu.

## 6.4 Nákupní seznam

Před samotnou implementací nákupního seznamu bylo důležité rozhodnout, jakou metodu ukládání dat bude aplikace využívat. Existuje několik možností, jak data ukládat, přičemž některé možnosti jsou lepší a jiné horší.

Do užšího rozhodování se rozhodně nedostaly možnosti jako použití SQLite databáze nebo Poskytovatele obsahu (ContentProvider), a to ani nemluvě o použití webového úložiště. Důvody pro tato rozhodnutí jsou vcelku zřejmé (časté připojování k databázi by mohlo způsobit zpomalení aplikace, poskytovatelé obsahu se používají především pro sdílení dat mezi aplikacemi, což v Receptáři rozhodně není zapotřebí, a navíc aplikace funguje, až na sdílení receptů, v režimu offline, tedy nepotřebuje k provozu připojení k Internetu. Proč to měnit?).

### 6.4.1 Shared Preferences vs. Internal Storage vs. External Storage

Jelikož aplikace podporuje vytvoření pouze jednoho nákupního seznamu a nikoli více seznamů (například pro každý obchodní řetězec jeden), bylo nejjednodušší variantou použití SharedPreferences. Důvody k tomuto rozhodnutí jsou popsány níže [14].

Aplikace by mohla využívat možnosti ukládání dat do interní paměti (Internal Storage) nebo do externí paměti (External Storage). Při vybrání externí paměti by však mohlo docházet k problémům. Například při odpojení této paměti může docházet k nekonzistentnosti dat a také samozřejmě k takto uloženým datům má přístup jakákoliv jiná aplikace i uživatel sám, a tudíž by se mohlo stát, že zbytečně dojde ke smazání uložených dat.

Při výběru mezi `SharedPreferences` a `InternalStorage` bylo nejdůležitějším faktorem velikost ukládaných dat. Jelikož nákupní seznam nebude obsahovat dlouhé texty ani nebude zabírat velký datový prostor, bylo jednodušší volbou použití `SharedPreferences`.

Třída `SharedPreferences` poskytuje obecný framework, který umožňuje ukládat a načítat trvalé data, která jsou uložena ve stylu klíč-hodnota. `SharedPreferences` mohou být použita pro ukládání jakýchkoliv primitivních dat, jako jsou například boolean hodnoty, celá čísla nebo řetězce, což jsou přesně ta data, o která se jedná při ukládání nákupního seznamu.

Přístup k těmto datům může být vcelku frekventovaný, například pokud uživatel při nákupu často přepíná mezi funkcemi aplikace. Toto by při použití `InternalStorage` mohlo způsobovat zpomalení aplikace častým zápisem a čtením ze souboru.

Uložená data pomocí `SharedPreferences` budou přetrvávat v uživatelských relacích, a to i když dojde k ukončení aplikace. Navíc k uloženým `SharedPreferences` má přístup pouze ta aplikace, která daná data vytvoří a uloží.

## 6.4.2 Uložení nákupního seznamu

Pro samotné uložení aktuální verze nákupního seznamu uživatel nemusí nikde hledat a potvrzovat ukládání. Jediné, co musí udělat, je stisknout tlačítko zpět. Aplikace je implementována tak, aby při ukončení aktivity uložila aktuální verzi do vybraných `SharedPreferences`.

Uživatel má možnost přidání nových položek do nákupního seznamu pouhým stiskem ikony v zobrazených surovinách receptu. V tomto případě nedochází k otevření aktivity Nákupní seznam, ale aplikace pouze pošle data do seznamu a ten je automaticky aktualizován a uložen. Uživatele tedy nic neruší v procházení receptem. Pokud by si chtěl zkontrolovat, co je již v nákupním seznamu uvedeno, má v menu rychlou volbu „Zobrazit nákupní seznam“.

Při následném otevření nákupního seznamu se data automaticky načtou z uložených `SharedPreferences` a zobrazí se v nákupním seznamu, kde je uživatel může jakkoliv editovat.

```
private void loadSavedPreferences() {
    SharedPreferences sharedPreferences =
    PreferenceManager.getDefaultSharedPreferences(this);
    int sizeofSharedPref = sharedPreferences.getAll().size();
    String value;
    for (int i = 0; i <= sizeofSharedPref+1; i++){
        value = sharedPreferences.getString(String.valueOf(i), "");
        if (value == "" || value == " "){
```

```

        //doNothing
    }else {
        pridatPolozkuDoSeznamu(value);
    }
}

private void savePreferences(String key, String value) {
    SharedPreferences sharedPreferences =
    PreferenceManager.getDefaultSharedPreferences(this);
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putString(key, value);
    editor.commit();
}

```

*Zdrojový kód 6: Načtení a uložení dat – Shared Preferences*

## 6.5 Časovač

Jednou z mnoha součástí aplikace Receptář je Časovač neboli Minutka. Pomocí časovače si uživatel může časovat výrobní proces receptu a samozřejmě funkcí časovače je upozornění na vypršení zadané doby.

Uživatel má možnost spuštění většího počtu časovačů zároveň a jejich přehled může vidět jak ve stavovém řádku (aktuálně běžící časovače), tak v aktivitě „Přehled časovačů“, kterou je možné vyvolat z menu hlavní aktivity. V přehledu časovačů následně uživatel vidí jak aktuálně běžící časovače, tak ty, které jsou vypnuté.

V této aktivitě je také samozřejmě implementována možnost vytvoření nového časovače, který se následně přidá do seznamu a uživatel jej může spustit.

Každý z těchto časovačů je možné také jednoduše pozastavit a později znovu spustit nebo úplně zastavit a spustit znova od začátku. Uživatel může také snadno kterýkoliv (neběžící) časovač smazat pouhým klikem na intuitivní ikonku popelnice, která je u každého časovače.

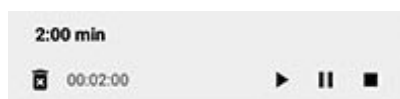
U časovačů lze také nastavit určité globální nastavení, které bude pro všechny časovače stejné. Tyto nastavení jsou následující. Uživatel si může nastavit tón vyzvánění při vypršení stanovené doby, stejně jako si může nastavit, jestli bude telefon vibrovat či nikoliv. Hlavním nastavením je však možnost tzv. nekonečného vyzvánění, kdy telefon upozorňuje na vypršení stanovené doby, dokud toto upozornění uživatel sám nevypne. Druhou možností je přehrání upozorňovacího tónu pouze jednou.

Tyto možnosti nastavení jsou řešeny pomocí `PreferenceActivity` [15], a tedy jsou ukládány automaticky při změně do `SharedPreferences`. Při každém spuštění dané aktivity pak dochází k načtení takto uložených nastavení.

### 6.5.1 Vzhled seznamu časovačů

Seznam časovačů je řešen pomocí upraveného `ListView`, kdy každá položka má tohoto seznamu má vlastní, upravený, vzhled. V každém řádku seznamu jsou uvedeny veškeré informace o časovači, jako je například název (který se skládá z počtu minut) a tlačítka s možnostmi spuštění, pozastavení a samotné zastavení časovače. Položka seznamu obsahuje také tlačítko pro smazání daného časovače. Takto upravená položka seznamu obsahuje samozřejmě také `TextView` s časem, který se po spuštění časovače aktualizuje každou vteřinu.

Samostatná položka seznamu vypadá následovně.



Obrázek 16: Položka seznamu

Ve výchozím stavu aplikace, kdy časovače netikají, je možné časovač pouze spustit, anebo smazat. Ovšem jakmile uživatel spustí časovač, je možnost smazání zrušena, a to až do doby, než časovač doběhne do konce nebo než je zastaven uživatelem. V době, kdy je časovač spuštěn má uživatel možnost pouze časovač pozastavit, nebo úplně zastavit.

### 6.5.2 Řešení časování výrobního postupu

Po spuštění časovače aplikace vytvoří novou instanci `AlarmManageru` [16], jehož funkcionality je popsána níže, a také se začne každou vteřinu aktualizovat daný `TextView`, který zastupuje časovač. Tato aktualizace je řešena vcelku jednoduše, a to pomocí nového vlákna rozhraní `Runnable`.

### 6.5.3 Upozornění - AlarmManager

Pro případ, kdy uživatel minimalizuje aplikaci je časovač upraven tak, aby zůstal běžící a tedy i upozornil uživatele na vypršení stanovené doby. Této funkce je docíleno pomocí `AlarmManageru` [16].

`Alarm Manager` je třída, která poskytuje přístup k službám alarmu zařízení. Toto umožňuje naplánování spuštění určité aplikace nebo její části v určitém okamžiku v budoucnosti. V závislosti na nastavení instance této třídy může alarm probudit zařízení i v případě, že je zařízení spící. Tato třída umožňuje mimo jiné také implementaci tzv. `PendingIntent`, ve kterém je možné zobrazovat průběh odpočtu ve stavovém řádku.

Použití třídy `Alarm Manager` bývá často svázáno se spouštěním určité akce pravidelně, například každý den ve stejnou dobu, k čemuž je využito metody `setRepeating()`, této možnosti však v Časovači není využito.

V následující části implementace je zobrazeno základní nastavení notifikace (upozornění), které se zobrazí ve stavovém řádku. Samotná notifikace je vytvořena pomocí třídy `NotificationCompat.Builder`, kde použitím metody `build()` je získána instance `Notification`. Ta je později zobrazena pomocí `NotificationManageru`.



```
final Notification notification = new NotificationCompat.Builder(this)
    .setSmallIcon(R.mipmap.ic_stat_onesignal_default)
    .setContentTitle("Kuchařka")
    .setContentText("Běží odpočet času...")
    .setContentIntent(pendingIntent).build();
```

*Zdrojový kód 7: Upozornění ve stavovém řádku*

K dalšímu nastavení notifikace dochází v pozdějších částech implementace, jako je například nastavení vibrací nebo vyzvánění. V případě, kdy si uživatel zvolí v nastavení možnost nekonečného přehrávání upozornění na vypršení stanovené doby, je toto v implementaci ošetřeno nastavením vlajky (*flag*) na `FLAG_INSISTENT` nebo `FLAG_ONLY_ALERT_ONCE`.

K takovému nastavení dochází pomocí následující ukázky implementace.

```
builder = new NotificationCompat.Builder(MyService.this);
SharedPreferences SP =
    PreferenceManager.getDefaultSharedPreferences(getBaseContext());
boolean vibrace = SP.getBoolean("vibrovat", true);
boolean nekonecnePrehravani = SP.getBoolean("nekonecnePrehravani",
false);
String vyzvaneni = SP.getString("vyzvaneni",
    "content://settings/system/notification_sound");
if (vibrace == true) {
    long[] pattern = {500, 500, 500, 500, 500, 500, 500, 500, 500};
    builder.setVibrate(pattern);
}
builder.setStyle(new NotificationCompat.InboxStyle());
Uri alarmSound = Uri.parse(vyzvaneni);
builder.setSound(alarmSound);
Notification myNotification = builder.build();
if (nekonecnePrehravani == true) {
    myNotification.flags |= Notification.FLAG_INSISTENT;
    builder.setAutoCancel(true);
} else {
    myNotification.flags |= Notification.FLAG_AUTO_CANCEL |
        Notification.FLAG_ONLY_ALERT_ONCE;
}
builder.setSound(alarmSound);
manager = (NotificationManager)
    getSystemService(Context.NOTIFICATION_SERVICE);
manager.notify(1, myNotification);
```

*Zdrojový kód 8: Nastavení možností časovače*

Styl upozornění se samozřejmě mění v závislosti na použitém nastavení zařízení, tedy pokud má uživatel nastaven tichý režim nebo vibrace, bude také upozornění na vypršení stanovené doby pouze vibrační, nikoli zvukové.

## 6.6 Rozšířené vyhledávání receptů

Rozšířené vyhledávání receptů podle zadané suroviny pracuje na jednoduchém principu. Aplikace vrací pouze ty recepty, které opravdu obsahují zadané suroviny. Jestliže uživatel zadá například 4 suroviny, z nichž 3 jsou v některém z receptů, ale ta zbývající ne, aplikace tyto recepty nezobrazí mezi vyhledanými recepty.

Další z funkcionalit při vyhledávání receptů je plně funkční našeptávač surovin, které jsou obsaženy v receptech. Uživateli stačí zadat první dvě písmena suroviny a následně se mu zobrazí seznam možných surovin, ze kterého si může uživatel vybrat namísto vypisování celého názvu suroviny. Pokud uživateli našeptávač nenapoví správnou surovinu, podle které chce vyhledávat, znamená to, že tuto surovinu žádný recept neobsahuje. Uživatel tedy musí zkusit zadat surovinu s jiným názvem.

## 6.7 Sdílení receptů

Pokud uživatel aplikace z menu vybere možnost sdílení receptu, aplikace mu nabídne seznam dostupných aplikací, pomocí kterých může sdílet daný obsah. Pro tuto nabídku dostupných aplikací je využívána akce `ACTION_SEND` [17] a jelikož uživatel sdílí pouze text, nikoli obrázky receptů, používá aplikace při posílání informací možnost `EXTRA_TEXT`.

```
Intent chooserIntent = Intent.createChooser(intentShareList.remove(0),
    "Sdílej recept pomocí ...");
chooserIntent.putExtra(Intent.EXTRA_INITIAL_INTENTS,
    intentShareList.toArray(new Parcelable[] {}));
startActivity(chooserIntent);
```

*Zdrojový kód 9: Výběr aplikace pro sdílení receptu*

Uvedená část zdrojového kódu provádí právě výše popsanou nabídku dostupných aplikací. Jelikož ale existuje velké množství aplikací, které umí pracovat s prostým textem, byla zapotřebí určitá filtrace, která by zaručila výběr pouze těch aplikací, které jsou určeny ke sdílení obsahu. Z tohoto důvodu byly přidány následující podmínky pro filtraci aplikací pomocí názvu balíku (*packageName*).

```
if (packageName.contains("com.facebook") ||
    packageName.contains("com.twitter.android") ||
    packageName.contains("com.google.android.apps.plus") ||
    packageName.contains("mms") ||
    packageName.contains("com.google.android.talk") ||
    packageName.contains("com.google.android.gm"))
```

```

{
    Intent intent = new Intent();
    intent.setComponent(new ComponentName(packageName, name));
    intent.setAction(Intent.ACTION_SEND);
    intent.setType("text/plain");
    intent.putExtra(Intent.EXTRA_SUBJECT, "Recept - " + nazev_receptu);
    intent.putExtra(Intent.EXTRA_TEXT, nazev_receptu + "\n\n" +
        ingredientsOfRecipe(recept) + "\n\n" + contentOfRecipe(recept) +
        "\n\n" + evaluationOfRecipe(recept));
    intentShareList.add(intent);
}

```

*Zdrojový kód 10: Filtrace výběru aplikací pro sdílení receptu*

V konečné verzi je možné recept sdílet pomocí aplikací Hangouts, Google+, Gmail, Messenger od Facebooku, Twitteru a SMS zprávy. Takovýto seznam aplikací uživatel uvidí samozřejmě pouze tehdy, jsou-li v telefonu nainstalovány. V případě, kdy uživatel nemá ani jednu ze zmíněných aplikací nainstalovanou, vypíše se chybová hláška („Nenalezeny žádné aplikace, pomocí kterých lze recept sdílet.“).

Jakmile si uživatel vybere, pomocí které aplikace bude obsah receptu sdílet, posílá Receptář data získané z databáze z daného receptu do vybrané aplikace. O zbylou funkcionalitu se již dále stará uživatelem vybraná aplikace.

## 7 Závěr

Cílem Bakalářské práce bylo vytvořit funkční aplikaci pro správu kuchařských receptů a jejich automatizované úpravy. Aplikace obsahuje možnost sdílení receptů pomocí základních aplikací, které uživatelé systému Android používají. Uživatelé aplikace mají také možnost dávat receptům své hodnocení a komentáře. I tyto informace jsou v případě sdílení receptu sdíleny spolu s veškerými informacemi, které daný recept obsahuje. V aplikaci funguje i tzv. „skenování spižírny“, kdy uživatel nemusí vyhledávat recepty jen podle názvu receptu, ale má také možnost zadávat do vyhledávače suroviny, které recepty obsahují. Navíc uživatel tyto suroviny nemusí jen ručně vypisovat, ale může také využít funkce skenování čárového kódu suroviny. Aplikace poté sama nabídne recepty, ve kterých se dané suroviny objevují.

V aplikaci byla také zabudována možnost vytvoření zcela nového receptu a také úpravy již existujících receptů, a to ať se jedná jen o jednoduché přepočítávání množství surovin podle počtu porcí, anebo celková úprava receptu, jako je například úprava ingrediencí nebo pracovního postupu. Dále aplikace umožňuje export potravin do editovatelného nákupního seznamu pro snadnější nákupy a také časování výrobního postupu se samozřejmým upozorněním na vypršení stanovené doby. Časovač výrobního postupu umožňuje také spuštění většího počtu samostatných časovačů. Aplikace tedy umožňuje uživateli vařit více jídel zároveň, a to beze strachu z toho, že zapomene na hotovou část receptu v troubě.

Stanovené cíle, které byly uvedeny v úvodu práce, byly úspěšně dosaženy, a to dokonce s určitými vylepšeními. V průběhu implementace aplikace bylo nalezeno několik málo možností zlepšení, a to jak vzhledu a orientace v aplikaci, tak přidání dalších funkcionalit, které původně ani nebyly v zadání. Jednou z důležitých změn vzhledu aplikace v průběhu implementace bylo vytvoření posuvných karet v zobrazení receptu. Tohoto bylo využito namísto nutnosti rolování v jedné velké aktivitě.

Každodenním problémem aplikace byla nedokonalá práce s databází, která musela být vyřešena. V průběhu implementace bylo nutné pozměnit přístup k databázi tak, aby byla dodržena konzistentnost dat. Tohoto bylo dosaženo díky implementace návrhového vzoru Jedináček, tedy použití přístupu k databázi pomocí jedné instance. Oproti předchozím implementacím byla aplikace, a tedy i databáze, v průběhu finálního testování velmi stabilní.

Závěrem bych ráda zmínila možnosti vylepšení aplikace, které jsou následující. V aplikaci by mohla být v pozdější době přidána možnost sdílení receptu i s obrázkem (samozřejmě pro případ, že obrázek je k receptu přiložen), což ve stávající verzi implementováno není. Dále by bylo možné vymyslet a použít určité koeficienty pro přepočet některých potravin, které by byly využívány při změně počtu porcí v receptu. Toto by se týkalo surovin, jako je například olej nebo máslo. Prozatím je přepočet počtu porcí v receptu řešen pouhým násobením popsáním v kapitole týkající se modifikací receptů. V neposlední řadě by také stála za zmínku možnost přidání sdíleného receptu do aplikace pomocí poslání JSON nebo XML souboru jakožto přílohy ve vybrané aplikaci pro sdílení. Další možnost vylepšení aplikace je samozřejmě doplnění dalších receptů do existujícího základního setu receptů v databázi, což je zdoluhavá práce a vcelku nedůležitá součást fungující aplikace – alespoň z pohledu programátora. S tímto také souvisí možnost přidání výživových hodnot k receptům, případně také předpřipravené jídelníčky na

několik dní nebo i týdnů zdravé stravy anebo detoxikace organismu. Do aplikace by mohly být zařazeny i určité články a doporučení ke zdravému životnímu stylu.

## Reference

- [1] Obchod Google Play. [online]. [cit. 2017-03-25].  
*Dostupné z:* <https://play.google.com/store/apps>
- [2] GENERAL RECIPE. sRecepty. [online]. [cit. 2017-03-20].  
*Dostupné z:* <https://play.google.com/store/apps/details?id=com.generalrecipe.srecipes>
- [3] GRUNCL, Jan. Česká kuchařka. [online]. [cit. 2017-03-20].  
*Dostupné z:* <https://play.google.com/store/apps/details?id=cz.jhg.ceskakucharka>
- [4] EL TORO.CZ. Miluji Vaření – recepty. [online]. [cit. 2017-03-20].  
*Dostupné z:* <https://play.google.com/store/apps/details?id=cz.eltoro.milujivareni.app>
- [5] ECONOMIA. Vaření.cz – Nejlepší recepty. [online]. [cit. 2017-03-20].  
*Dostupné z:* <https://play.google.com/store/apps/details?id=cz.pesek.vareni>
- [6] TEAM, ZXing. Barcode Scanner. [online]. [cit. 2017-03-20].  
*Dostupné z:* <https://play.google.com/store/apps/details?id=com.google.zxing.client.android&hl=cs>
- [7] ČÁPKA, David. UML - Use Case Diagram. [online]. [cit. 2017-03-28].  
*Dostupné z:* <http://www.itnetwork.cz/navrhove-vzory/uml/uml-use-case-diagram>
- [8] ČÁPKA, David. UML - Use Case Specifikace. [online]. [cit. 2017-03-28].  
*Dostupné z:* <http://www.itnetwork.cz/navrhove-vzory/uml/uml-use-case-specifikace-diagram>
- [9] PASTORČÁK, Petr. Sekvenční diagram. [online]. [cit. 2017-03-28].  
*Dostupné z:* <http://orca.xf.cz/ooms/010/010.htm>
- [10] ČÁPKA, David. UML - Class Diagram. [online]. [cit. 2017-04-06].  
*Dostupné z:* <http://www.itnetwork.cz/navrhove-vzory/uml/uml-class-diagram-tridni-model>
- [11] ČÁPKA, David. Singleton (jedináček). [online]. [cit. 2017-03-25].  
*Dostupné z:* <http://www.itnetwork.cz/navrhove-vzory/singleton-navrhovy-vzor>
- [12] AsyncTask. [online]. [cit. 2017-03-26].  
*Dostupné z:* <https://developer.android.com/reference/android/os/AsyncTask.html>
- [13] KONEČNÝ, Matěj. Vyvíjíme pro Android: Notifikace, broadcast receivery a Internet. [online]. [cit. 2017-03-26].  
*Dostupné z:* <https://www.zdrojak.cz/clanky/vyvijime-pro-android-notifikace-broadcast-receivery-a-internet/>

- [14] Storage Options. [online]. [cit. 2017-03-21].  
*Dostupné z:* <https://developer.android.com/guide/topics/data/data-storage.html>
- [15] PreferenceActivity. [online]. [cit. 2017-03-28].  
*Dostupné z:* <https://developer.android.com/reference/android/preference/PreferenceActivity.html>
- [16] AlarmManager. [online]. [cit. 2017-03-28].  
*Dostupné z:* <https://developer.android.com/reference/android/app/AlarmManager.html>
- [17] Sending Simple Data to Other Apps. [online]. [cit. 2017-03-28].  
*Dostupné z:* <https://developer.android.com/training/sharing/send.html>
- [18] MISHA. Databázové modely. [online]. [cit. 2017-04-06].  
*Dostupné z:* <http://www.databaze.chytrak.cz/modely.htm>

# Seznam příloh

Obsah přiloženého CD ROM:

\CookBook – kompletní zdrojové kódy aplikace

\Android – soubor pro instalaci hry

\CIZ0031.pdf – textová část bakalářské práce ve formátu PDF